What is claimed is:

1   1.   A communication method comprising the steps of:

2   providing a first executable application at a server, the first executable application

3   requiring a first amount of memory for proper execution;

4   executing a second executable application at a client, the second executable

5   application occupying a second amount of memory;

6   providing a first data portion at a client, the first data portion occupying a third

7   amount of memory, the first data portion being accessible by the second executable

8   application;

9   providing a second data portion at a client, the second data portion having a lower

10   priority than the first data portion, the second data portion occupying a fourth amount of

11   memory, the second data portion being accessible by the second executable application;

12   receiving an input at the client, the input corresponding to the first executable

13   application on the server; and

14   determining, in response to receiving the input, whether the client has sufficient

15   available memory for execution of the first executable application.

1   2.   The method of claim 1, wherein the step of providing the first data portion

2   further comprises the step of:

3   providing a current program data file.

1   3.   The method of claim 2, wherein the step of providing the second data

2   portion further comprises the step of:

3   providing a future program data file.

1   4.   The method of claim 1, further comprising the step of:

2   downloading the first executable application in response to determining that the

3   client has sufficient available memory for execution of the first executable application.

1   5.   The method of claim 1, further comprising the steps of:

2   purging the second data portion from the memory in response to determining that

3   the client has insufficient available memory for execution of the first executable

4   application; and

5   deciding, in response to purging the second data portion, whether the client has

6   sufficient available memory for execution of the first executable application.

1   6.   The method of claim 5, further comprising the step of:

2   downloading the first executable application in response to deciding that the client

3   has sufficient available memory for execution of the first executable application.

1   7.   The method of claim 5, further comprising the steps of:

2   purging the first data portion from the memory in response to further determining

3   that the client has insufficient available memory for execution of the first executable

4   application; and

5   assessing, in response to purging the first data portion, whether the client has

6   sufficient available memory for execution of the first executable application.

1   8.   The method of claim 7, further comprising the step of:

2   downloading the first executable application in response to assessing that the

3   client has sufficient available memory for execution of the first executable application.

1   9.   The method of claim 7, further comprising the steps of:

2   purging the second executable application from the memory in response to

3   assessing that the client has insufficient available memory for execution of the first

4   executable application; and

5   evaluating, in response to purging the second executable application, whether the

6   client has sufficient available memory for execution of the first executable application.

1   10.   The method of claim 9, further comprising the step of:

2   downloading the first executable application in response to evaluating that the

3   client has sufficient available memory for execution of the first executable application.

1   11.   A communication system comprising:

2   a client having a tuner, the client being capable of receiving broadcast

3   information, the client being in bi-directional communication with a server, the server

4   having an executable application, the executable application requiring an amount of

5   memory for proper execution; and

6   a memory manager in the client, the memory manager being configured to

7   determine whether the client has sufficient available memory for proper execution of the

8   executable application, the server being configured to transmit the executable application

9   in response to the memory manager determining that the client has sufficient memory for

10   proper execution of the executable application.

1   12.   A communication method comprising the steps of:

2   establishing a bi-directional communication pathway between a client and a

3   server, the client having a tuner, the client being capable of receiving broadcast

4   information, the server having an executable application, the executable application

5   requiring an amount of memory for proper execution;

6   receiving an input at the client, the input corresponding to the executable

7   application on the server; and

8   determining, in response to receiving the input, whether the client has sufficient

9   available memory for proper execution of the executable application.

1   13.   The method of claim 12, wherein the memory is a volatile memory.

1   14.   The method of claim 12, further comprising the steps of:

2   allocating the required amount of memory from the available memory in response

3   to determining that the client has sufficient available memory for proper execution of the

4   executable application; and

5   requesting the executable application from the server upon allocating the required

6   amount of memory.

1   15.   The method of claim 12, further comprising the steps of:

2   purging data contained in the memory in response to determining that the client

3   has insufficient available memory for proper execution of the executable application; and

4   determining, in response to purging the data, whether the client has sufficient

5   available memory for proper execution of the executable application.

1   16.   The method of claim 15, further comprising the steps of:

2   purging a pre-existing application contained in the memory in response to

3 determining that the client has insufficient available memory for proper execution of the

4 executable application after purging data contained in the memory; and

5   determining, in response to purging the pre-existing application, whether the client

6 has sufficient available memory for proper execution of the executable application.

1   17.   The method of claim 15, wherein the step of purging data comprises the

2 step of:

3   purging the memory in accordance with a dynamic list of priorities.

1   18.   The method of claim 15, wherein the step of purging the memory

2 comprises the steps of:

3   purging a data file having television viewing information for future days prior to

4 purging a data file having television viewing information for a current day; and

5   purging a data file having television viewing information for a current day prior to

6 purging an application loaded upon initialization.

1   19.   The method of claim 15, further comprising the steps of:

2   iteratively repeating the purging and determining steps until the client has

3   sufficient available memory for proper execution of the executable application;

4   allocating the required amount of memory from the available memory in response

5   to determining that the client has sufficient available memory for proper execution of the

6   executable application; and

7   requesting the executable application from the server upon allocating the required

8   amount of memory.

1   20.   The method of claim 15, wherein the step of purging the data comprises

2   the step of purging data in accordance with a dynamic list of priorities.

1   21.   The method of claim 12, further comprising the steps of:

2   purging a pre-existing application contained in the memory in response to

3   determining that the client has insufficient available memory for proper execution of the

4   executable application; and

5   determining, in response to purging the pre-existing application, whether the client

6   has sufficient available memory for proper execution of the executable application.

1   22.   The method of claim 21, further comprising the steps of:

2   iteratively repeating the purging and determining steps until the client has

3   sufficient available memory for proper execution of the executable application;

4   allocating the required amount of memory from the available memory in response

5   to determining that the client has sufficient available memory for proper execution of the

6   executable application; and

7   requesting the executable application from the server upon allocating the required

8   amount of memory.

1   23.   The method of claim 21, wherein the step of purging the pre-existing

2   application comprises the step of purging the memory in accordance with a dynamic list

3   of priorities.

1   24.   The method of claim 12, further comprising the steps of:

2   compacting a private heap in response to determining that the client has

3   insufficient available memory for proper execution of the executable application; and

4   determining, in response to compacting the private heap, whether the client has

5   sufficient available memory for proper execution of the executable application.

1   25.   The method of claim 12, further comprising the steps of:

2   compacting a system heap in response to determining that the client has

3   insufficient available memory for proper execution of the executable application; and

4   determining, in response to compacting the system heap, whether the client has

5   sufficient available memory for proper execution of the executable application.

1   26.   A communication method comprising the steps of:

2   establishing a bi-directional communication pathway between a client and a

3   server, the client having a tuner, the client being capable of receiving broadcast

4   information, the server having an executable application, the executable application being

5   executable using a normal amount of memory, the executable application further being

6   executable using a reduced amount of memory;

7   receiving an input at the client, the input corresponding to the executable

8   application on the server; and

9   determining, in response to receiving the input, whether the client has sufficient

10   available memory for execution of the executable application using the normal amount of

11   memory.

1   27.   The method of claim 26, further comprising the steps of:

2   allocating the normal amount of memory from the available memory in response

3   to determining that the client has sufficient available memory for execution of the

4   executable application using the normal amount of memory; and

5   requesting the executable application from the server upon allocating the normal

6   amount of memory.

1   28.   The method of claim 26, further comprising the steps of:

2   determining, in response to determining that the client has insufficient available

3   memory for execution of the executable application using the normal amount of memory,

4   whether the client has sufficient available memory for execution of the executable

5   application using the reduced amount of memory.

1   29.   The method of claim 28, further comprising the steps of:

2   allocating the reduced amount of memory from the available memory in response

3   to determining that the client has sufficient available memory for execution of the

4   executable application using the reduced amount of memory; and

5   requesting the executable application from the server upon allocating the reduced

6   amount of memory.

1   30.   The method of claim 28, further comprising the steps of:

2   purging data contained in the memory in response to determining that the client

3   has insufficient available memory for execution of the executable application using the

4   reduced amount of memory; and

5   determining, in response to purging the data, whether the client has sufficient

6   available memory for execution of the executable application using the reduced amount

7   of memory.

1   31.   The method of claim 30, further comprising the steps of:

2   iteratively repeating the purging and determining steps until the client has

3   sufficient available memory for execution of the executable application using the reduced

4   amount of memory;

5   allocating the reduced amount of memory from the available memory in response

6   to determining that the client has sufficient available memory for execution of the

7   executable application using the reduced amount of memory; and

8   requesting the executable application from the server upon allocating the reduced

9   amount of memory.

1   32.   The method of claim 30, wherein the step of purging the data comprises

2   the step of purging data contained in memory in accordance with a dynamic list of

3   priorities.

1   33.   The method of claim 28, further comprising the steps of:

2   compacting a private heap in response to determining that the client has

3   insufficient available memory for execution of the executable application using the

4   reduced amount of memory; and

5   determining, in response to compacting the private heap, whether the client has

6   sufficient available memory for execution of the executable application using the reduced

7   amount of memory.

1   34.   The method of claim 28, further comprising the steps of:

2   purging a pre-existing application contained in the memory in response to

3   determining that the client has insufficient available memory for execution of the

4   executable application using the reduced amount of memory; and

5   determining, in response to purging the pre-existing application, whether the client

6   has sufficient available memory for execution of the executable application using the

7   reduced amount of memory.

1   35.   The method of claim 34, further comprising the steps of:

2   iteratively repeating the purging and determining steps until the client has

3 sufficient available memory for execution of the executable application using the reduced

4 amount of memory;

5   allocating the reduced amount of memory from the available memory in response

6 to determining that the client has sufficient available memory for execution of the

7 executable application using the reduced amount of memory; and

8   requesting the executable application from the server upon allocating the reduced

9 amount of memory.

1   36.   The method of claim 34, wherein the step of purging the pre-existing

2 application comprises the step of purging the pre-existing application in accordance with

3 a dynamic list of priorities.

1   37.   The method of claim 28, further comprising the steps of:

2   compacting a system heap in response to determining that the client has

3 insufficient available memory for execution of the executable application using the

4 reduced amount of memory; and

5   determining, in response to compacting the system heap, whether the client has

6 sufficient available memory for execution of the executable application using the reduced

7 amount of memory.

1   38.   A communication method comprising the steps of:

2   establishing a bi-directional communication pathway between a client and a

3   server, the client having a tuner, the client being capable of receiving broadcast

4   information, the server having an executable application, the executable application

5   requiring an amount of memory for proper execution, the server being configured to

6   retransmit data over sequential time intervals;

7   receiving an input at the client, the input corresponding to the executable

8   application on the server; and

9   determining, in response to receiving the input, whether the client has sufficient

10   available memory for execution of the executable application.